

24
150 pág.

ARUN

Enciclopedia Práctica del Spectrum



Nueva Lente/Ingelek



EN MOVIMIENTO

GOMD ya vimos en el capítulo anterior, se conocen como sprites los móviles empleados en los programas, generalmente presentes en juegos de acción, pero también en educativos o programas de cualquier otro tipo. Los sprites son unidades gráficas que pueden componerse de un sólo carácter, pero usualmente lo están por un grupo de ellos que se desplazan conjuntamente. Esto es posible gracias a que los matrices de 8×8 , que componen los caracteres del Spectrum, en base a los cuales se construyen los móviles, no dejan ninguna separación ni vertical ni horizontal entre ellos, de este modo, pueden aplicarse caracteres definidos unas encima de otros, o de forma lateral, para conseguir un sprite de mayor tamaño.

En este mismo capítulo, encontraremos un juego de acción, en el cual será premiado nuestro puntaje a la hora de abatir aviones enemigos, que sobrevuelan negligentemente nuestras fronteras aéreas.



Los sprites son unidades gráficas dotadas de movimiento, los cuales pueden estar compuestas por uno o más caracteres.

El programa se basa en los conocimientos ya adquiridos de como generar caracteres del usuario, producir efectos de color, y otros muchos, pero entre ellos se hace uso de una sentencia nueva **INKEY%**. La importancia de esta nueva función, merece que le dedicaremos una explicación pro-

va, antes de entrar al comentario general de su funcionamiento en el programa.

INKEY%

Básicamente, la función **INKEY%** nos permite obtener el último carácter pulsado en el teclado, y utilizar dicho valor de cualquier forma que deseemos.

Así pues, esta función nos devuelve un sólo carácter, pero que pueda ser almacenado en una variable, impreso en la pantalla, etc. dado que en su forma de función, la flexibilidad para el ma-

!

Dado que las matrices de 8×8 que configuran los caracteres no presentan ninguna separación al ser impresas, conjuntamente en la pantalla, podemos generar unidades gráficas continuas en base a caracteres simples.

*

INKEY%, como cualquier otra función, debe ir precedida de una palabra clave, que indique la forma de aplicar el resultado.

*

INKEY% permite averiguar el último carácter pulsado en el teclado.

nejo del resultado es enorme, y depende directamente de la palabra clave que, obligatoriamente, debe anteponerse a la propia función.

Hasta ahora, y puesto que ya conocemos la existencia de **INPUT**, esta función no nos parece muy importante, sino más bien un caso excesivamente restringido de dicha sentencia. No obstante, aunque ambas sentencias estén destinadas a la toma de datos, la diferencia entre ellas es sustancial. **INKEY%** no detiene la ejecución del programa.

Si la toma de datos del programa hubiera sido

!!

INKEY\$ no considera caracteres significativos, de manera independiente, ninguno de los dos SHIFT, CAPS o SYMBOL.

*

INKEY\$ consulta el teclado en el momento de su ejecución, pero no detiene el programa.

realizado por INPUT, el objetivo de nuestros dispositivos se iría deteniendo cada vez que avanzara un carácter en la pantalla y, de esa manera, sería evidentemente sencilla lanzar un proyectil en el momento adecuado.

Así pues, siempre que deseemos que una acción continúe, ya sea el movimiento de un sprite o cualquier otra, y que al mismo tiempo se efectúe una toma de datos, es imprescindible el uso de la función INKEY\$.

Lógicamente, dos cosas no pueden suceder al mismo tiempo, y si nos encontramos desplazando el móvil, no estaremos realizando la toma de datos, sin embargo, las cosas se sucederán a tal velocidad en el programa, que tendremos una sensación de simultaneidad.

Ahora bien, puesto que dado eso alta velocidad de la que hemos hablado no podemos calcular en qué momento pasa el programa por la sentencia INKEY\$, ¿qué sucederá si en el momento de su ejecución no existe ninguna tecla pulsada? La respuesta es bien fácil, la función nos lo hará saber tomando por resultado la cadena nula (vacía). Debido a esta facultad, la función INKEY\$ se suele emplear para que se efectúe una detención del programa hasta que pulsemos alguna tecla. Por ejemplo: `10 IF INKEY$="" THEN GO TO 10.`

Como podemos observar, el programa continuará ejecutando el bucle de la línea 10 siempre y



Mientras que INPUT detiene la ejecución de las programadas, INKEY\$ efectúa la entrada de datos sin producir más efectos.

La función INKEY\$ nos permite averiguar el carácter pulsado en el teclado.



cuando el resultado de INKEY\$ sea la cadena nula, o dicho en otras palabras, mientras que no exista ninguna tecla pulsada.

Bien es cierto, que la instrucción PAUSE 0 conseguiría un efecto parecido, y con menos ocupación de espacio, sin embargo, éste sólo será similar. Tengamos en cuenta que PAUSE 0 se limita a aguardar la pulsación de cualquier tecla, mientras que INKEY\$ utilizado en la estructura anteriormente estudiada, en combinación con IF, además de aguardar la pulsación de una tecla, informa de cuál ha sido el carácter entrado.

UTILIDAD DE INKEY\$

Si su uso es prácticamente obligado en los MENUS, término con el cual se designan los puntos del programa en los cuales éste se detiene en espera de que el usuario seleccione una entre varias opciones. Observemos esto sobre un ejemplo práctico:



```

10 GLS
20 INVERSE 1
30 PRINT "----- MENU DE OPCIONES -----"
40 PRINT "'1' INVERSE 0:" Opcion
uno
50 PRINT "'2' INVERSE 0:" Opcion
dos
60 PRINT "'3' INVERSE 0:" Opcion
tres
70 INVERSE 0
80 IF INKEYS="1" THEN PRINT "SELEC-
CIONADA OPCION UNO" STOP
90 IF INKEYS="2" THEN PRINT "SELEC-
CIONADA OPCION DOS" STOP
100 IF INKEYS="3" THEN PRINT "SELEC-
CIONADA OPCION TRES" STOP
110 GO TO 80

```

En esta ocasión, se han combinado tres estructuras **IF INKEYS**, una detrás de otra, por un sistema que se conoce como **CASCADA**, de forma que si ninguna de las tres se cumple, es decir, tanto si no se ha pulsado ninguna tecla, como si la tecla no se encontraba dentro del rango de las opciones, se volverá a realizar la toma de datos. Al pasar, el hecho de que **INKEYS** realice la entrada de un sólo carácter, pero la ventaja además de no necesitar la pulsación de **ENTER**, y por tanto, no producir la detención del programa simplifica enormemente las tareas de depuración.



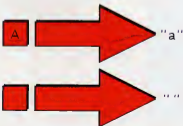
La función **INKEYS** se hace prácticamente imprescindible en los juegos de acción.

se efectúa una depuración previa de los datos, como podría ser **80 IF INKEYS<"1" OR INKEYS>"3" THEN GO TO 80**.

Sin embargo, cuando se utiliza **INKEYS** varias veces seguidas, cabe la posibilidad de que, si ha transcurrido un cierto tiempo desde la depuración de la entrada hasta el análisis de la misma, ésta se haya alterado, con el correspondiente error. Veamos un ejemplo.

El siguiente programa, va a multiplicar por 2 un número tomado mediante **INKEYS** y, por tanto, de un sólo dígito. El primer problema será realizar una depuración para que sólo se tomen valores numéricos, y acto seguido, convertir mediante **VAL** el resultado en forma de cadena devuelto por **INKEYS** a número tratable por el operador de multiplicación (*).

Cuando el operador **INKEYS** no existe ningún tecla pulsada, se devuelve como resultado la cadena vacía.



INKEYS se utiliza frecuentemente para la configuración de menús de programas.

De hecho, sobre todo cuando el número de opciones es más grande, no se suele utilizar el método expuesto de decisión en cascada, sino que



```
10 IF INKEY$="" OR INKEY$="9" THEN
GO TO 10
20 FOR I=0 TO 499: NEXT I
30 PRINT VAL INKEY$?2
```

Dado que **INKEY\$** efectúa la lectura de un solo carácter, no precisa de la pulsación posterior de **ENTER** como sucede con **INPUT**.



El uso de **INKEY\$** es prácticamente obligatorio en cualquier programa de escape.



La pulsación simultánea de **CAPS SHIFT** y **SYMBOL SHIFT (EXTENDED)** da que es crucial en cuenta por **INKEY\$** devolviendo el resultado **CHR\$ 54**.

El hecho de que el programa tarde en ejecutarse algún tiempo, se debe al retardo de la línea 20, que ha sido introducido intencionadamente, para producir una diferencia de tiempo entre el momento de la depuración de datos (línea 10), y el de su análisis (línea 30). A menos que hayamos operado pacientemente, con la tecla pulsada hasta que aparezca el resultado, el programa se detendrá con un mensaje **C No sense in BASIC**. ¿A qué es debido esto? ¿Acaso no ha funcionado el sistema de depuración?

En absoluto. El problema radica en que el valor de **INKEY\$** no permanece almacenado, sino que depende del momento de la ejecución de la función, y por tanto, es susceptible de cambios extraordinariamente colincas. Así, cuando se pulsó una tecla, la línea de depuración funcionó a la perfección y filtró cualquier código que no estuviera comprendido entre cero y nueve.

No obstante, al llegar al programa a la línea 30, el valor de **INKEY\$** había pasado a ser nulo, al cesar la pulsación de la tecla, y por tanto se produjo un error cuando intentó efectuarse el **VAL** sobre una cadena vacía.

Para evitar este tipo de inconvenientes, el sistema más frecuente de uso para **INKEY\$** es en combinación con la asignación a una variable (**LET**), de forma que el resultado de la función



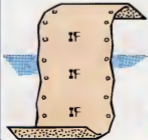
Con **INKEY\$** existe la posibilidad de que entre el punto de depuración de estado y el de análisis, el resultado se vea alterado.

quede registrado, y permanezca constante en todo momento desde la depuración hasta su análisis. Transformemos el ejemplo anterior

```
10 LET X$=INKEY$: IF X$="" OR
X$="9" THEN GO TO 10
20 FOR I=0 TO 499: NEXT I
30 PRINT VAL X$?2
```

Un último problema que surge con frecuencia al utilizar la función **INKEY\$** es el debido a su extraordinaria rapidez. Ejecutando el siguiente mini-programa, y pulsando cualquier tecla, nos damos cuenta de ello inmediatamente.

En los algoritmos es crucial, así **IF** sucede a otro



```
10 CLS
20 PRINT INKEY$:
30 GO TO 20
```

Para evitar este efecto de autorreplicación, en muchas ocasiones no deseado, utilizamos una estructura en base a dos **INKEY\$**, que facilitan la toma de datos mediante esta función, pero carácter e carácter. Veámos una aplicación práctica de éste en una nueva versión del último mini-programa

```
10 CLS
20 IF INKEY$="" THEN GO TO 20
30 PRINT INKEY$:
40 GO TO 20
```

En la línea 20 se generan los gráficos correspondientes a la A, B y C, a partir de la configuración decimal expuesta en la DATA de la línea 400. En este caso, al ser tres caracteres consecutivos los que componen el gráfico, podemos permitirnos el «ahorro» de efectuar 24 POKEs seguidos, comenzando en la dirección de inicio dada por USR "A".

Esto nos hace llegar a la conclusión de que expresiones tales como USR "A"+8 y USR "B" son sinnónimas a USR "A"+16 y USR "C". Efectivamente, puesto que USR "<letra>" no constituye más que la forma abreviada de expresar una dirección, podemos permitirnos utilizarla en la forma que nos convenga, como si de cualquier otro valor numérico se tratara.

Las líneas 30 a 70 efectúan la inicialización general del juego, es decir, ajustan los marcadores a los valores iniciales, dibujan los terrenos anteriores, etc.

Como podemos observar, el artificio del cambio es la línea 20, la cual actúa como filtro e impide el paso a la impresión hasta que no se ha dejado de pulsar la tecla. La aplicación más concreta de este método, se da con una doble estructura IF INKEY\$, la primera de las cuales elimina el efecto de autorrepresión, y la segunda propicia la espera de la pulsación de cualquier tecla.

```
10 IF INKEY$="" THEN GO TO 10
20 IF INKEY$="" THEN THE GO TO 20
30 PRINT INKEY$
40 GO TO 10
```



La forma más segura de escribir INKEY\$ es en combinación con la constante LET, para abaratar el resultado de la función.



INKEY\$ tiene en cuenta el modo de cursor L o C, para devolver el carácter pulsado en mayúsculas o minúsculas.



Tengamos siempre en cuenta que el resultado devuelto por INKEY\$ es una cadena.

multilínea de los dos SHIFT, CAPS y SYMBOL, generando el resultado CHR\$ 14, correspondiente al código de separador de número en el código ASCII del Spectrum, pero que también se utiliza para la representación de EXTENDED MODE.

Cabe destacar que INKEY\$ se ve igualmente afectado por los modos C y L del cursor, de forma que el resultado de la función en modo C, al

La gran utilidad de operación de INKEY\$ puede observarse en ciertos momentos de un procedimiento.

RESULTADO DE INKEY\$

Para finalizar con este tema, mencionaré cuál es el resultado que devuelve INKEY\$. Como era de suponer, la función no se ve afectada por todas las teclas, sino que ignora las pulsaciones de aquellas que no equivalen a ningún carácter, como es el caso de SYMBOL SHIFT o CAPS SHIFT.

Sin embargo, sí los tiene en cuenta para la composición de caracteres que las precisan, como es el caso de los símbolos que se obtienen mediante SYMBOL SHIFT o las mayúsculas, obtenidas pulsando simultáneamente CAPS SHIFT.

Así mismo, también se advierte la pulsación si-



!!

Delante la expresión de INKEYS, la pulsación de CAPS SHIFT junto con una tecla alfabética, permite la obtención de las mayúsculas.

*

La pulsación de SYMBOL SHIFT simultáneamente a una tecla alfanumérica nos permite el acceso a los símbolos de las teclas desde la función INKEYS.

*

Por demás se conoce el conjunto de aparatos que en determinados momentos de un programa se lo presentan al usuario.

pulsar cualquier letra, será siempre la mayúscula, sin necesidad de concurrir la pulsación de CAPS SHIFT.

Así pues, es importante hacer notar que, como dijimos al comenzar el estudio de este función, su resultado es el carácter pulsado en el momento de la ejecución y no la tecla, dado que en ocasiones ambas expresiones no son sinónimas. Por ejemplo, con aquellos caracteres que precisan para su obtención de la pulsación de un SHIFT

COMIENZA LA ACCION

Como lo prometido es deuda, he aquí el esperado programa anunciado al comienzo de este capítulo, que aplica a un juego la función INKEYS, y pone en práctica nuestros conocimientos recientemente adquiridos sobre los atributos de color y la generación de gráficos definibles.

```
10 REM ANTIADREO - J.H. LOPEZ MARTINEZ
20 FOR I=0 TO 23: READ X: POKE USR "A"
+1,X: NEXT I
30 @BORDER 5: PAPER 5: LET R=0
40 CLS : PRINT AT 21,0: PAPER 61: @BORDER
@BORDER @BORDER @BORDER @BORDER @BORDER
50 PRINT AT 20,0: @BORDER 15: "222" : TA
B 25: "231"
60 PRINT AT 0,0: BRIGHT 1: INK 1: "PLANT
ON...": B: BRIGHT 0: BRIGHT 1: "NUNCIOS!":
50: BRIGHT 0
70 LET D=0: LET P=0: LET H=0
80 LET A=INT (USR41412)
90 FOR L=0 TO 20
100 PRINT AT ALL: INK 2: "ABC"
110 IF 0 THEN GO TO 150
120 LET X=INKEYS: IF XK<" " OR X=0:3'
THEN GO TO 200
130 LET B=10*VAL X-4: LET D=D+1: LET H=H
-1: LET H=59
140 PRINT AT 0,27-LEN STR$ B: BRIGHT 1:
" : B
150 LET H=H-1
160 IF H<0: THEN GO TO 100
170 IF B%2 AND B%4=0 THEN GO TO 260
180 PRINT AT H,B: " : AT H+1,B: "
190 IF H%2 THEN GO TO 200
200 LET S=B: PRINT AT H,B: "
210 IF NOT H THEN GO TO 320
220 GO TO 200
230 FOR T=0 TO 3: NEXT T
240 NEXT L
250 PRINT AT A,29: " : GO TO 80
260 PRINT AT H+1,B: " : AT A,L: INK 2: P
LASH 1: "BOM"
270 FOR T=0 TO 99: NEXT T
280 LET D=0: PRINT AT A,L: "
290 LET P=16+A%4*(4-(10-C1)/10+1)
300 PRINT AT A,12-LEN STR$ P: BRIGHT 1:
P
310 IF H THEN GO TO 80
320 IF P<4R THEN GO TO 350
```

```
330 PRINT AT 0,0: "HAS ESTABLECIDO UN NU
OVO RECORD!"
340 LET R=P: GO TO 360
350 PRINT AT 0,9: "EL RECORD SIOLE"
360 PRINT AT 11,16-(10+LEN STR$ R)/2: "E
N T: BRIGHT 1:R: BRIGHT 0: " PUNTOS"
370 PRINT AT 13,2: "DESEA INTENTARLO DE
NUEVO?"
380 LET X=INKEYS: IF X=0:3' OR X=0:3'
THEN GO TO 90
390 IF XK<"N" AND XK<"Y" THEN GO TO 10
300
400 DATA 0,190,220,300,255,159,127,0,0,
0,7,15,255,193,255,0,0,0,120,193,240,255
,240,0
```

La estructura general del programa anterior es de una estructura muy simplificada.



A partir de este momento, comenzamos a hacer uso del color, asignando tanto al fondo de la pantalla (PAPER) como al marco (BORDER) un color azul celeste, con el cual simulamos el despegado cata, escenario de la aeronave batista.

La línea 40 incorpora un CLS que hace extensivo el atributo de color PAPER a toda la pantalla. En este caso, las sentencias de color presentes en la línea 30 y la de la 40 (CLS) se han superado, cosa aparentemente ilógica. Esto se debe a que 40 marca la división entre los inicializadores de programa y los de juego.

Esto quiere decir que las líneas de la 10 a la 30 sólo son ejecutadas una vez, al lanzarse por primera vez el programa. Estas son las llamadas inicializaciones de programa, y fijan valores que permanezcan inalterables hasta finalizar el mis-



$$E \equiv \text{CHR}\$ 14$$

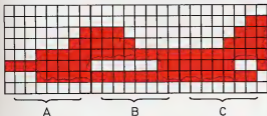
Para INKEY\$ el cursor E (EXTENDED) equivale a un CHR\$ 14.

INKEY\$



La pulsación independiente de CAPS SHIFT y SYMBOL SHIFT es ignorada por INKEY\$.

El gráfco abstrcto en el programa anterior está formado por los gráfcos definidos de las teclas A, B y C.



mo, tales como las definiciones de gráficos, los colores principales del juego (fondo y marco), y el record, almacenado en la variable R.

En esta zona de programa se encuentran todos los gráficos predefinidos. Así pues, los caracteres doblamente subrayados de las líneas 40 y 50 deben ser introducidos como los gráficos combinados de las teclas afectadas; es decir, los que se obtienen por la pulsación simultánea de CAPS SHIFT y la tecla correspondiente en el modo GRAPHICS (CAPS SHIFT + B).

En la línea 80 se genera de forma aleatoria la altura del avión para cada nueva pasada por la pantalla. Como es evidente, se le han marcado unos límites, de forma que no vuele tan alto como para borrar a su paso los marcadores de la línea superior, ni tan bajo que sea demasiado fácil de derribar o chocar con los posiciones de los antiaéreos.

Es en este punto en el cual comienza lo que podríamos llamar ciclo principal del programa, puesto que es dentro de él en el cual discurre el juego, hasta que se termina la misión.



Entre el punto de pulsación de la entrada por INKEY\$ y su entrada puede que se produzca una alteración del resultado de la función.



Para evitar posibles errores por variaciones de resultado al sacar sistemas para utilizar INKEY\$ es en combinación con una sentencia de asignación LET.

!

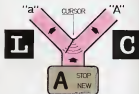
Una estructura de decisión en cascada se aplica en la cual los IF se suceden, según merda bifurcaciones por distintos condiciones.

La línea 90 marca el comienzo del bucle FOR NEXT de la variable L, en el cual se comprende el desplazamiento horizontal del avión, y que tiene su fin en el NEXT de la línea 240. Esta es la zona de programa que más frecuentemente se ejecuta, y contiene la entrada y tratamiento de los datos, así como la rutina de desplazamiento del avión.

En las líneas 100 y 110, se imprime el avión en su posición correspondiente y se envía, por medio de la variable D, si hay algún proyectil en el aire. Si existe alguno, no se permite disparar otro (sería un abuso), por lo cual el programa se bifurca a la línea 150.

Tengamos en cuenta a la hora de introducir el programa, que el sprite del avión está construido con caracteres gráficos, y por tanto, las letras ABC subrayadas, que aparecen en la línea 100 (impresión del avión), deben ser sustituidas por los caracteres gráficos de las teclas afectadas. En la línea 120 se aceptan caracteres del teclado por medio de la sentencia INKEY\$. Como los dígitos se producen pulsando las teclas del 1, 2 y 3, está claro que cualquier otro valor, inclu-

Las líneas 160 a 230 del programa contienen sus las comparas de colisión si el proyectil ha hecho blanco.



El resultado de INKEY\$ se ve afectado por los botones C y L del cursor.

yendo el que no se haya pulsado ninguna tecla (X\$="") no es válido, forzando el programa al final del bucle de L para proseguir el desplazamiento del avión.

Fijémonos en que la desviación se dirige concretamente a la línea 230, que sirve de retardo al programa, empleando un bucle FOR-NEXT vacío, para ajustar la diferente velocidad de los datos de programa en los cuáles no se pulsan teclas válidas, puesto que se efectúan en este caso menos comparaciones con el consiguiente incremento en la velocidad de ejecución.

En las líneas 130 a 150 se calcula la posible puntuación (B), en proporción a la batería que efectúa el disparo, se conecta el indicador de proyectil en el aire (D), se resta un proyectil, y se establece el valor inicial de desplazamiento vertical de este, actualizándose el marcador de munición.

En las líneas 160 y 230, se verifica si ha existido colisión entre el proyectil y el avión, comparando los valores de las ordenadas y abscisas respectivas. Si no se produce la colisión, se continúa con la impresión del proyectil, de existir el choque, se bifurca a la línea 260, fuera del bucle donde se trata la explosión.

En las líneas 300 y 360, se imprime el acumulado de puntos, y se comprueba si aún queda munición, de ser así, se vuelve a poner en circulación un nuevo avión. Si se han agotado los proyectiles, se prosigue con el análisis de la puntuación, para comprobar si se ha establecido un nuevo record.

Las líneas 370 a 390 contienen la evaluación de la respuesta al mensaje de si se desea jugar una nueva partida. Por último, la línea 400 contiene la DATA donde se almacenan los patrones del sprite utilizado para el avión (gráficos de la A, B y C).



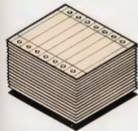
CASETE SI, CASETE NO



L almacenamiento masivo de datos ha constituido de por sí uno de los caballos de batalla más traídos y llevados entre los fabricantes de micros en los últimos años

algunas deficiencias del sistema, pero virtudes como también otras muchas virtudes lo hacen especialmente útil, incluso insustituible, en función del trabajo que deseamos realizar

A FAVOR Y EN CONTRA



Por el menor precio de una C-40 podemos almacenar de un mismo fichero unos 600 Kb. de información.

Tres son los factores primordiales que condicionan la utilidad de cualquier periférico como soporte exterior de información:

1. Tiempo de acceso a la información
2. Tiempo de transmisión de ésta desde o hacia el ordenador
3. Fiabilidad del proceso.

Obviamente, el mejor sistema sería aquel que consiguiera localizar la información casi instan-

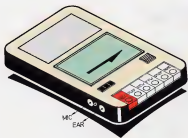
El cassette es el medio de almacenamiento masivo más utilizado en los microordenadores.

te. No sólo por la guerra de software que desencadena, sino por todas aquellas cuestiones referentes a rapidez, fiabilidad y precio, que van pareja al soporte utilizado como medio de almacenamiento exterior de la información. Sin duda, el más popular de todos, es el cassette.

Quizás, por parte de los profesionales de la informática, el mero hecho de mencionar en su presencia la palabra cassette, haga aparecer una marea de horror o desagrado en sus rostros. Incluso un pequeño ataque de nervios puede llevar a la desesperación al técnico, que por exigencia del trabajo, necesita de acceso y transmisión de datos a elevadas velocidades, si tan sólo dispone de un cassette de audio convencional para efectuar su labor.

Pero los usuarios del Spectrum, al menos en una etapa inicial, ponemos mucha más confianza en las posibilidades que como medio de acopio masivo de datos y programas nos puede proporcionar el cassette. No obstante, hemos de reconocer





Las portátiles «cinta del tipo «compacta» suelen dar muy buenos resultados con un bajo precio.

técnicamente, enviarla o recibirla en el menor tiempo posible, y además, no dejara lugar a errores de transmisión (máxima fiabilidad). Desgraciadamente para nosotros, disponer de un equipo que conjugue óptimamente las tres posibilidades, conduce directamente a un cuarto factor, tan importante como los tres anteriores: el precio. Este será tanto más elevado, cuanto más aumenten la velocidad y seguridad en los procesos de transferencia de datos que involucran a nuestro porfínico encargado de gestionar el tránsito de la información.

Precisamente el precio es una de las mayores virtudes de los casetes. Por un lado, no necesitaremos un modernísimo equipo de alta fidelidad para poder usarlo con nuestro Spectrum. Bastará una pequeña grabadora portátil, a la cual exigiremos un mínimo de cualidades que comentaremos más adelante. En la actualidad, por menos de diez mil pesetas, pueden conseguirse en el mercado aparatos que cumplen a la perfección su cometido. Por otro lado, el medio utilizado como soporte de la información, lo que popularmente se conoce como cinta casete, es con diferencia el más barato entre los utilizados para el almacenamiento masivo de datos. Basta decir como ejemplo, que una cinta C-60, de una hora de duración (treinta minutos por cada cara), con un precio medio en torno a las doscientas pesetas, es capaz de almacenar del orden de 640 Kbytes.



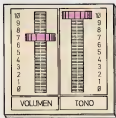
Las pilas suelen gastar más rápido en el momento más importante, conviene que nuestro casete disponga de un sistema de almacenamiento por red.

VELOCIDAD Y FIABILIDAD

Muchos de los usuarios se quejan de la extrema lentitud, cuando se trata de grabar o recuperar programas almacenados en cinta. Bien, es cierto, y además, estos tiempos de espera pueden llegar a exasperar al más paciente. Estudiemos algo más a fondo este problema.

Técnicamente, en la mayoría de los sistemas informáticos, la medida utilizada para cuantificar la velocidad de transmisión de los datos, a través de una línea de comunicaciones o de conexión entre periféricos, es el bit por segundo. Esta unidad recibe el nombre de Baudio. Por tanto, 1 baudio es equivalente a 1 bit/segundo.

Los fabricantes de nuestro micro fijaron esta velocidad a 1500 baudios, aproximadamente. Y aunque seguramente preguntará: ¿y por que no una más elevada?



Si el casete no dispone de opciones automáticas de volumen y tono, es conveniente ajustar el primero a dos cuartos de la escala, y el segundo al máximo de aquélla.

La respuesta es muy sencilla: es física. Es decir, asegura que el ordenador no se hará un lío al interpretar la información que recibe desde la grabadora, ni que la cinta magnética llegue a saturarse, de manera que los datos allí almacenados resulten irreconocibles, para el Spectrum, durante el proceso de carga.

No obstante, las rutinas que gestionan todo lo relacionado con la grabación y carga de datos en casete, están albergadas en la R.O.M., es decir,

el control del proceso se realiza por software. Desde luego, el contenido de la memoria de sólo lectura (R.O.M.), como sabemos, es inalterable, a menos que nos empeñemos, por medios más o menos violentos, y nada aconsejables, en tratar de modificarla.

Obtenido con cierta habilidad, y utilizando técnicas en absoluto perjudiciales para la integridad física de la R.O.M. del Sistema, veremos en un próximo capítulo como la velocidad de transmisión puede ser elevada a 2000, 2500, 3000 baudios e incluso más. O podremos disminuirla, con el sano objetivo de obtener copias de seguridad, con una calidad elevada.

Segará habiendo quien piense que el casete es lento, comparado con otros periféricos empleados en el almacenamiento masivo de información, pero quedará demostrado que no tanto como la mala prensa le atribuye.

ACCESO EN CASETE

Cuestión muy diferente es la relacionada con el acceso a un determinado programa o bloque de datos contenido dentro de una cinta. No queda otro remedio que realizarlo de forma secuencial, avanzando y retrocediendo la cinta, actuando sobre las teclas adecuadas, hasta localizar el fragmento deseado. Todo ello contribuye a elevar el tiempo de acceso, sobre todo si se trata de cintas de larga duración.

En la actualidad, aparecen en el mercado algunos casetes especiales que llevan incorporado un sistema de búsqueda rápida de programas, lo cual supone una considerable ventaja, si tomamos en cuenta que lo normal es tener almacenados varios bloques de información dentro de la misma cinta.

Es precisamente este factor el que los detractores de los casetes argumentan, no exentos de razón, como «falta de operatividad». Consideremos, por ejemplo, un fichero en el cual tuviéramos almacenados el nombre de nuestros amigos y sus respectivos números de teléfono. Si alguna de nuestras amistades cambia de número, no tendríamos más remedio que extraer de la cinta el fichero completo, modificar en la memoria el contenido de la variable que conserva el número amigo, y volver a grabar en cinta el nuevo fichero.

Por tanto, la posibilidad de disponer de ficheros de acceso directo, los cuales permiten la modificación o lectura de un dato específico, sin necesi-



Un mal alineamiento de la cabeza de lectura/escritura (bobinas) puede ser la causa de problemas en la lectura de programas grabados con otras cintas o algunos por causas naturales.

TORNILLO REGULADOR



Las bobinas que poseerán todo el relacionado con la grabación y la carga de datos en casete, están albergadas en la R.O.M. es decir, el control de estos procesos se realiza por el firmware.



En la actualidad aparecen en el mercado algunos casetes especiales que incorporan sistemas de búsqueda automática de programas.

El mecanismo de avance de la bobina es muy importante para la lectura/escritura de programas.



!!

Estos son los factores principales que condicionan la utilidad de cualquier periférico como soporte externo de información:

1. Tiempo de acceso a la información.
2. Tiempo de transición de esta desde o hacia el ordenador.
3. Fiabilidad del proceso.

*

El precio es uno de los mayores factores de los casetes. No necesitamos un modernísimo equipo de alta fidelidad (basta una pequeña grabadora portátil

este, e adoptemos el método a seguir para solventar el problema.

En un futuro, no muy lejano, el abaratamiento de otros sistemas como el disco o las memorias masivas a base de burbujas magnéticas, que permite una rápida transferencia y localización de los datos, desplazará al cassette, relegándole a un segundo término, donde su cometido no será otro que el de conservar copias de seguridad de los programas y ficheros de uso corriente. Hasta entonces, la relación precio-capacidad, lo convierten en el periférico más utilizado como medio de almacenamiento de memoria en masa.

UNA GRABADORA IDEAL

Los onerosos acórrimos del uso del cassette como soporte de información, basan sus quejas en los excesivos problemas y elevado porcentaje de errores que pueden producirse durante los labores de lectura/escritura.

En efecto, estos son los momentos críticos, pero la elección adecuada, tanto de la cinta como de la grabadora, puede disminuir el tanto por ciento de error a niveles bajísimos.

La idea de seleccionar nuestro Spectrum a un modernísimo equipo de alta fidelidad, debe ser desestimada. No sólo por problemas de espacio, sino también porque el elevado número de mecanismos destinados a filtrar la señal, atenuarla, equalizarla, disminuir el nivel de ruido o seleccionar las frecuencias adecuadas para la occu-



No cree usted de que la velocidad de transferencia del cassette depende de que desate.

rra, tienden a complicar en exceso el manejo. Y todo ello, sin contar con el factor precio, lógicamente más elevado en estos equipos.

Los pequeños casetes portátiles, ya sea del tipo utilizado por los periodistas o cualquier otro modelo, suelen ofrecer las suficientes prestaciones para trabajar con nuestro micro. Dentro de éstos, la elección debería ir encaminada hacia uno mono, frente al estéreo. Si bien los últimos ofrecen mejores resultados cuando se trata de reproducir música, la utilización de dos pistas durante el proceso de grabación (grabación estéreo), tiende a provocar problemas de reproducción al menor desajuste de los cabezales de lectura/escritura.

Además del botón PLAY, empleado para la lectura, y del RECORD o REC, que pulsado junto con el anterior, pone el aparato en situación de escritura, serán necesarios como mínimo otros tres pulsadores. Uno de STOP, para detener la grabadora, y otros dos para facilitar el avance (F-FWD, Fast ForWard) y rebobinado (RWD, ReWind) rápido de la cinta. Algunos modelos, permiten el uso conjunto de PLAY+FFWD y PLAY+RWD, con el objeto de controlar mediante el altavoz o los auriculares, la diferencia de sonido entre las zonas grabadas y en blanco, y así detectar el posible comienzo de un programa.

La tecla de pausa no es del todo necesaria, aunque sí muy útil. Permite detener la marcha de la grabadora durante el trabajo de lectura/escritura, sin necesidad de estar constantemente parando y volviendo a poner en marcha el aparato con el uso de PLAY y STOP. El control de tono también puede ser accesorio, y cese de tenerlo, debemos situarlo hacia el máximo de agudos.

El aumento en la velocidad de transferencia conduce a una disminución de la fiabilidad de la grabación.



Resulta imprescindible un sistema de ajuste del volumen durante las operaciones de lectura. Desgraciadamente, no todas las cassetes están grabadas al mismo volumen, y será preciso actuar sobre este control para ajustar el adecuado (normalmente, hacia los tres cuartos del máximo). Por lo general, los grabadores portátiles incorporan control automático del nivel de grabación, que resulta normalmente adecuado a los cognoscas de nuestro Spectrum. Actualmente, los denominados «cassetes especiales para ordenador», suelen incorporar un sistema automático de volumen, tanto para la lectura como para la grabación.



Las cintas de acceso directo están voladas al cassette

El cuantivuelitas de cinta constituye una inestimable ayuda a la hora de buscar con mayor rapidez un programa determinado. Conservar las cintas ordenadas, indicando en cada una de ellas los programas que contiene, junto con los datos referentes a longitud, dirección de comienzo (ver programa INDEX) y número de vuelta en el que está ubicado, nos ahorrará innecesarias pérdidas de tiempo al tratar de localizar un programa. El uso de pilas en grabadoras portátiles está generalizado. Pero es muy importante que el aparato disponga de una salida para poder conectar directamente a la red, a través de una pequeña fuente de alimentación. La experiencia de nuestra que las pilas tienen la mala costumbre de gastarse en el momento más importante, ocasionando irregularidades en el desplazamiento de la cinta, y como consecuencia, problemas en la transmisión de los datos.

La salida EART y la entrada MIC deben tener el mismo diámetro (3,5 mm), que los clavijas de los cables suministrados con el Spectrum. Algunos aparatos llevan una tercera salida, denominada REMOTE o REM, utilizado para controlar desde el ordenador el motor del cassette, pero que con el Spectrum carece de utilidad.

El sistema de anastre debe ser de calidad. De otra manera, inevitablemente tendremos problemas desde el primer momento. La velocidad de desplazamiento de la cinta está normalizada a 4,76 centímetros por segundo. Por ello, lo más recomendable es adquirir un equipo de algún fabricante de contrastada calidad, y no dejarse engañar por la afirmación: «Cuanto peor sea el cassette, mejor cargará el Spectrum». Los cassetes de nueva aparición, que los fabricantes denominan especiales para ordenador, habitualmente no son otra cosa que aparatos normales dotados de volumen constante para lectura y escritura, y un dispositivo que permite mantener simultáneamente conectadas las clavijas EART y MIC. Tendremos que considerar si merece la pena pagar bastante más que por un equipo normal.

Sobre la cabeza de lectura/escritura, algunos cassetes poseen un pequeño orificio, bajo el cual se encuentra un tornillo, destinado al ajuste de la inclinación de la cabeza. Esta debe quedar perpendicular a la cinta, y es asombrosa la facilidad con que se desajusta, dando lugar a problemas durante la carga. Con la grabadora en posición PLAY, y utilizando una cinta de las que contienen información para el Spectrum, podemos girar lentamente el tornillo, mediante un destornillador adecuado, hasta escuchar el sonido lo más estridente y agudo posible.

Esta será la posición de ajuste adecuada, aunque si seguimos teniendo problemas por este motivo, lo mejor es acudir a un servicio técnico especializado. A partir de ese momento, la grabadora comenzará a funcionar mejor, pero, seguramente, los programas contenidos en otras cintas grabadas anteriormente, con los cabezas desajustadas, nos den algún problema a la hora de intentar recuperarlos.



La idea de conectar nuestro Spectrum a un ordenador es un equipo de alta fidelidad debe ser deseado.



!

Una cinta de una hora de duración es capaz de almacenar del orden de 640 Kb.

*

La medida utilizada para cuantificar la velocidad de transmisión de los datos es el Baud, y equivale a un bit por segundo. La velocidad media de transmisión del Spectrum son 1500 Baudios.

*

Utilizando discos en absoluto preparados para la ROM del Sistema, podemos elevar la velocidad de transmisión.

EL PROFESOR

LOS ordenadores ayudan al ser humano en multitud de campos, tales como el científico, comercial, estadístico, etc. En nuestro caso, los microordenadores, donde manifiestan una particular habilidad es en la enseñanza de las tareas automáticas, fundamentalmente a los más pequeños.

Son muchos los países que han adoptado este método de enseñanza: Japón, Estados Unidos, Inglaterra. La misma abierta de los jóvenes encuentra en el ordenador un amigo que los ayuda y aconseja en sus tareas. Además, y puesto que irremisiblemente estas máquinas -impersonales- son, con toda certeza, parte importante de su futuro, no está de más que desde muy temprana edad se acostumbren y familiaricen con ellas, y comprendan que los ordenadores no son más que aparatos móviles si no existe un humano que los maneje.

La programación orientada a la enseñanza es sumamente sencilla, aunque no exenta de alguna que otra complicación mínima. En nuestro caso, hemos realizado un pequeño programa, que sirve como botón de muestra de la inmensa cantidad de aplicaciones desarrolladas en esta área.

EL PROGRAMA

Nuestro programa se ocupa de un simplísimo problema aritmético: la suma. Este está limitado por el número de sumandos y la longitud de cada uno de ellos. La cantidad máxima de sumandos que admite el programa es diez, y la longitud extrema de cada uno, no ha de exceder de diecisiete dígitos.

El listado de este programa se podía haber reducido a unos pocos líneas, pero la gran cantidad de mensajes explicativos que se han incluido, con el fin de entablar un diálogo eficaz con el usuario, hacen que la extensión del programa sea algo más acortada. Cuanto menor es la edad del usuario, mayores deben de ser las precauciones que tome el programador para la correcta asimilación y comprensión de los conceptos que utiliza el programa.

Los programadores debemos tener siempre en cuenta, que un programa es tanto mejor, cuanto menos preciso del conocimiento informático del usuario. En ese sentido, los programas destinados a los jóvenes, más que ninguno, deben contemplar todas las posibles circunstancias de error que puedan concurrir, de forma que en ningún momento el joven usuario se vea despreciado o imposibilitado ante la máquina.

Así mismo, es conveniente que el ordenador estable un diálogo lo más ágil posible con el estudiante, de forma que éste llegue a considerarlo más como un compañero o un amigo mayor, que como una simple máquina. Este tipo de relación usuario-ordenador, se conoce como método CONVERSACIONAL, y debe de estar presente en todos los programas educativos como el que hoy presentamos, sufriendo punto por punto al destinatario del programa.

De no cumplirse estas normas, lo más seguro es que todos nuestros esfuerzos por crear un buen programa educativo resulten infructuosos, aunque hoyamos empleado una gran cantidad de tiempo y conocimientos en el componente técnico del mismo.

!

Cada uno de los sumandos que componen la operación de adición no ha de exceder de diecisiete dígitos.

*

Una vez introducidos todos, y cuando uno de las líneas de instrucción del programa pidiere al usuario un número, presione una tecla, para el siguiente comando SAVE "PROFESOR". Si este comando por la grabación son autoevaluación, teclearemos la instrucción con SAVE "PROFESOR" LINE 10.

*

Los programas educativos deben encontrarse automáticamente protegidos contra posibles circunstancias de error.

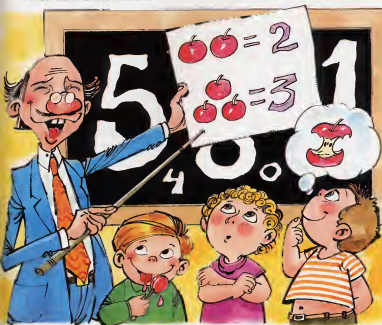
¡Guece a este programa, cambiamos le última línea por el listado del ordenador.



Como es habitual, para la grabación del programa utilizaremos el comando **SAVE**, con la forma **SAVE "PROFESOR"** o bien **SAVE "PROFESOR" LINE 10**, si deseamos que el programa se autoejecute nada más cargarlo.

Para la introducción del listado debemos tener en

cuenta que las palabras **INV** y **TRUE** subrayadas y entre corchetes, deben ser sustituidas, respectivamente, por los caracteres **INVERSE VIDEO (CAPS SHIFT + 4)** y **TRUE VIDEO (CAPS SHIFT + 3)**.



PROGRAMA

```

1 DEM *****
2 DEM * J.K.NAYRAL GERONDO *
3 DEM *****
4 DEM * PROFESOR 1900 *
5 DEM *****
18 PAPER 1: 18K % POKI 23058,8
28 NUMBER 1: CLS
31 LET PWP=60 LET E=9
32 LET FL=0
33 LET FILA=12: LET COL=0
34 LET O#="OLA" SOY TU AYUDANTE DE SUMAS: GO 258
398
31 PAUSE 50
32 PRINT AT 14,5: PAPER 2: QUIEN ERES TU ?
33 INPUT AS
34 CLS
71 LET COL=0
80 LET O#=" PERFECTO '+AS+'
91 LET PWP=3: LET FL=1
92 GO SUB 3000
83 FOR N=1 TO 30
84 KEEP -1,INT (RND*2)+60
85 NEXT N
86 GO SUB 3000
96 PRINT PAPER 5:AT 2,4: WAMOS A EMPEZAR
98 PRINT
110 PRINT "DIME CUANTOS SUMANDOS TIENE TU"
130 PRINT "PROBLEMA"
140 INPUT A
150 IF A=1 THEN GO TO 190
169 CLS
170 LET O#="ESTO ES MUY SERIO! NO BROMEES!"
171 LET FILA=20: LET COL=1
172 GO SUB 3000
173 PAUSE 50
180 CLS + GO TO 100
190 IF A<11 THEN GO TO 250
200 CLS
210 PRINT "LO SIENTO, NO PUEDO AYUDARTE EN"
220 PRINT AT 2,8:"UNA SUMA TAN LARGA"
230 PAUSE 250
240 GO TO 1110
250 CLS
259 LET B=8
270 DIM B$(A,20)
280 PRINT "DE ACUERDO "A#
290 PRINT
300 FOR I=1 TO A
310 PRINT "ESCRIBE EL SUMANDO "I
320 INPUT C#
330 IF C#="" THEN GO TO 320
340 IF LEN C#>20 THEN GO TO 390
350 CLS
360 PRINT "NO SE SUMAN CIFRAS TAN LARGAS"
370 PAUSE 200
380 GO TO 1110
390 FOR J=1 TO LEN C#
400 IF C$(J)="/" OR C$(J)="/" THEN GO TO 440
410 NEXT J
420 GO TO 490
430 PRINT
440 CLS
450 PRINT "SIN TRAMPAS: SOLO NUMEROS"
460 PRINT AT 2,8:"OTRA VEZ"
470 PRINT
480 GO TO 310
490 IF LEN C#>1 THEN GO TO 530
500 IF C$(1)="/" THEN GO TO 530
510 LET C#<C# TO J
520 GO TO 490
530 IF LEN C#>8 THEN LET B=LEN C#
540 LET C#=""
550 LET B$(1)=C#LEN C#-19 TO J
560 PRINT
570 NEXT I
580 CLS
590 PRINT "BUENO "J#
600 PRINT AT 2,8:"WAMOS A EMPEZAR LA SUMA"
610 PAUSE 150
620 CLS
630 FOR I=1 TO A
640 PRINT AT 1+8-A/2,15-9/2+I*(1,20-8 TO )
650 NEXT I
660 PRINT TAB 16-8/2:
670 FOR I=1 TO B
680 PRINT " "
690 NEXT I
700 LET C#=""
710 LET O#="17
720 FOR I=8 TO 2 STEP -1
730 LET O#="5/2+I
740 PRINT PAPER 6:AT 10+A/2,O#""
750 LET C#<I#KEYS
760 IF C#="/" AND C#="/" THEN GO TO 800
770 PRINT AT 10+A/2,O# "
780 LET C#<I#KEYS
790 IF C#="/" OR C#="/" THEN GO TO 780
800 PRINT AT 10+A/2,O#C#
810 GO SUB 1270
820 LET F=C+I#INT (E/10)
830 IF WA, C#<F THEN GO TO 800
840 PRINT AT 8,0:"MUY BIEN "J#"" SIGUE"
850 PAUSE 100
860 LET O#="E/10)
870 GO TO 780
880 PRINT AT 8,0:"NO "J#":, NO SON "J#
890 PAUSE 100
900 PAUSE 100
910 PRINT AT 8,0:
920 IF WA, C#<F THEN GO TO 780
930 NEXT I
940 PRINT AT 8,0:"VENGA "J#": YA ACABAMOS"
950 PRINT "CUAL ES EL ULTIMO NUMERO?"
960 INPUT "ENTER">,P
961 LET K#<I#KEYS
970 GO SUB 1270
980 IF O# THEN GO TO 1010
990 PRINT AT 8,0:"NO ES ESO, PRUEBA OTRA VEZ"
1000 GO TO 940
1010 LET C#<STR# F
1020 PRINT AT 10+A/2,0-LEN C#)C#
1030 PRINT AT 8,0:
1040 FOR J=0 TO 56
1050 PRINT " "
1060 NEXT J
1070 PRINT AT 8,0:"FANTASTICO "J#
1080 PRINT AT 2,9:"CUNDO HAYAS COPIAO PULSA "S""
1090 LET C#<I#KEYS
1100 IF C#>"S" THEN GO TO 1090
1110 CLS
1120 PRINT AT 8,0:"PROGRAMAS CON OTRA SUMA"
1130 INPUT C#
1140 IF C#>"S" THEN GO TO 1170
1150 CLS
1160 GO TO 70
1170 IF C#>"N" THEN GO TO 1220
1180 CLS
1190 PRINT AT 9,9-LEN A#/2:"HASTA OTRA "J#
1200 PRINT AT 11,4:"YA PUEDES DESCONECTARME"
1210 GO TO 10000
1220 PRINT "COMO QUE ""J#""
1230 PRINT "1230 RETURN
1240 PRINT "NO ENTENDO ESO, DIME SI O NO"
1240 GO TO 1130
1270 LET E=C
1280 FOR I=1 TO A
1290 IF B$(1,20-8+I)="" THEN LET B$(1,20-8+I)="/"
1300 LET E=C+VAL B$(1,20-8+I)
1310 NEXT J
1320 RETURN
3000 REM (INV_250R_)ESCRITURA(18)
2010 FOR P#1 TO LEN B#
3020 PRINT PAPER P#:INK K: FLASH FL:AT FILA,COL#F:
0(17)
3025 DEEP ,01,INT (RND*98)
3030 NEXT P#
3040 RETURN
3050 REM (INV_300R_)BORRADO ESCRITURA(18)
3060 FOR P#1 TO LEN B#
3070 PRINT PAPER 1: FLASH B:AT FILA,COL#F+1
3080 NEXT P#
3090 RETURN

```

ELECTRICIDAD Y ENERGIA

ENCICLOPEDIA PRACTICA

ENERGIA NUCLEAR • INSTALACIONES ELECTRICAS • ELECTRODOMESTICOS FUNDAMENTOS DE ELECTRICIDAD • ENERGIA SOLAR • LA POTENCIA CONTRATADA • EL RECIBO DE LA LUZ



MUY IMPORTANTE: TODAS AQUELLAS PERSONAS QUE FORMALICEN SU SUSCRIPCIÓN ANTES DEL 8 DE NOVIEMBRE DE 1978, RECIBIRÁN DE FORMA TOTALMENTE GRATUITA UN SENSACIONAL ESTUQUE DE HERRAMIENTAS VALORADO EN 3.000 PESETAS.

Oferta válida únicamente para España

NOMBRE _____ TOKO _____
 APELLIDOS _____
 DOMICILIO _____
 CIUDAD _____ PROVINCIA _____
 C. POSTAL _____ TELÉFONO _____ PROFESIÓN _____

- Deseo suscribirme a la obra **ELECTRICIDAD Y ENERGIA** (51 fascículos más 4 fascículos extra de **PRIMER FASCÍCULO**) al precio de 10.100 pesetas
- Deseo suscribirme a la obra **ELECTRICIDAD Y ENERGIA** (51 fascículos más 4 fascículos extra de **TERCER FASCÍCULO**) al precio de 9.100 pesetas
- Si yo y/o mi familia (o las personas mencionadas) puede recibir una suscripción gratuita indicando desde qué número deseo suscribirme y descontando 176 pesetas por cada uno de ellos por obtener el precio total de la suscripción
- Deseo suscribirme a la obra **ELECTRICIDAD Y ENERGIA** a partir del número _____ por el importe de _____ pesetas
- Recibo —que abonaré de su totalidad con el primer envío— la tarifa eléctrica de la siguiente forma:
- Cuentas —con abono del importe más gastos de envío
- Dinero postal: n.º _____
- Tarifa mensual: adjunto a nombre de **INTELEX S.A.**
- Tarifa **VISA**: n.º _____
- Tarifa **MASTER CARD**: n.º _____
- Fecha de caducidad de la oferta _____
- Nombre del titular de la oferta _____

IMPORTANTE: los envíos se realizarán en grupos de 4 fascículos

Envío

Presenta o recoge esta cupón y envíalo a: dentro de un sobre a Ediciones EVSELEK, S.A. Apartado de Correos 61284, 20028 MANDRI

SPECTRUM

5 CINTAS de

JUEGOS

AUN

OFERTA ESPECIAL
1.495 PTAS

Ingelek



YA ESTA A LA VENTA EN SU QUIOSCO